The Phoenix Project: Coordinated Flight of Multiple Unmanned Aerial Vehicles

David F. Lieb ’03 & Mark R. Siano ‘03

Princeton University

Professor Robert Stengel

FAA/NASA Joint University Program Quarterly Review

January 9, 2003 – Atlantic City, NJ

Long Term Goal

- Fleet of autonomous aircraft
- Waypoint navigation
- Coordinated aerobatic maneuvers
- Team-oriented decision making

Applications

Motivation for inexpensive, small-scale UAVs:

- Remote Sensing
- Reconnaissance
- Search and Rescue
- Communication Network
- Etc.

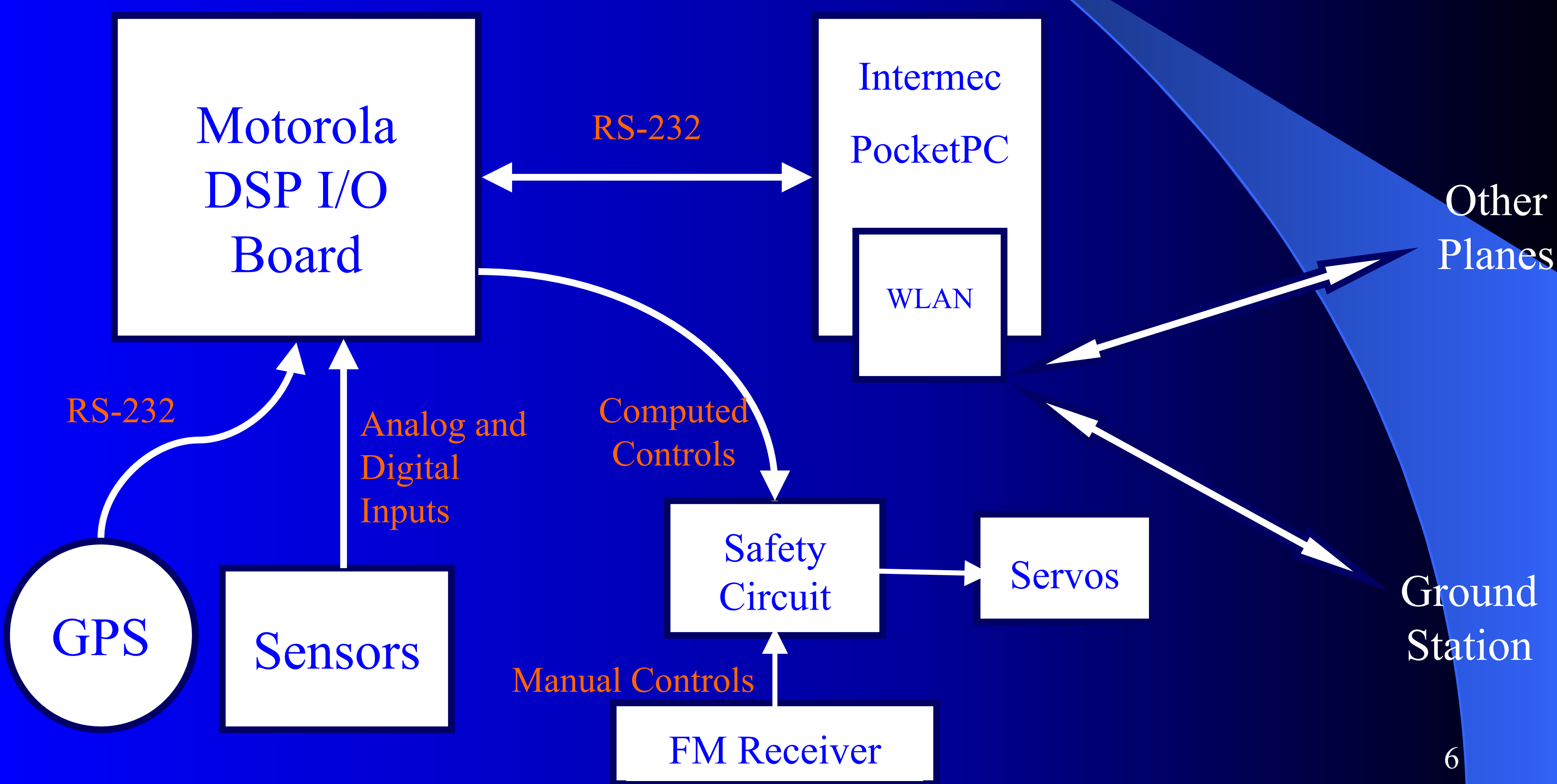
History

- Phoenix I – Data acquisition, modeling
  - Hobbico HobbiStar 60
  - TattleTale Data Logging Computer
  - Crash (human error)
- Phoenix II – Steady, level, autonomous flight
  - Added Cassio Cassiopeia as main processor
  - Integrated wireless LAN communication
  - Crash (prior to autonomous control)

Phoenix II - Problems

- TattleTale (16 MHz) could not stay in real-time
  - GPS data lagged system
- Control computations performed on Cassiopeia
  - Non-modular
  - Excess communications
- Bad wiring…

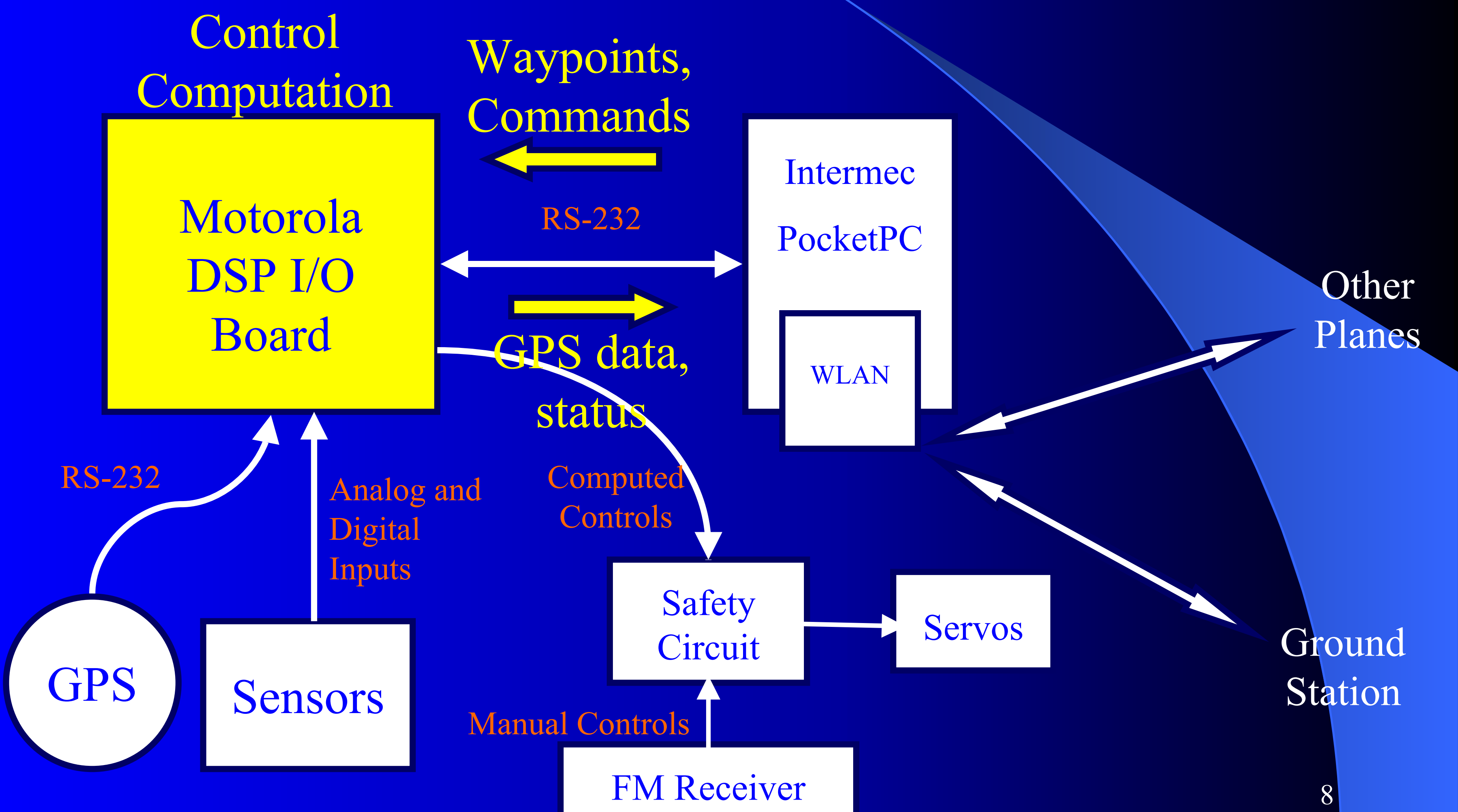
System Redesign



Tower Trainer 60

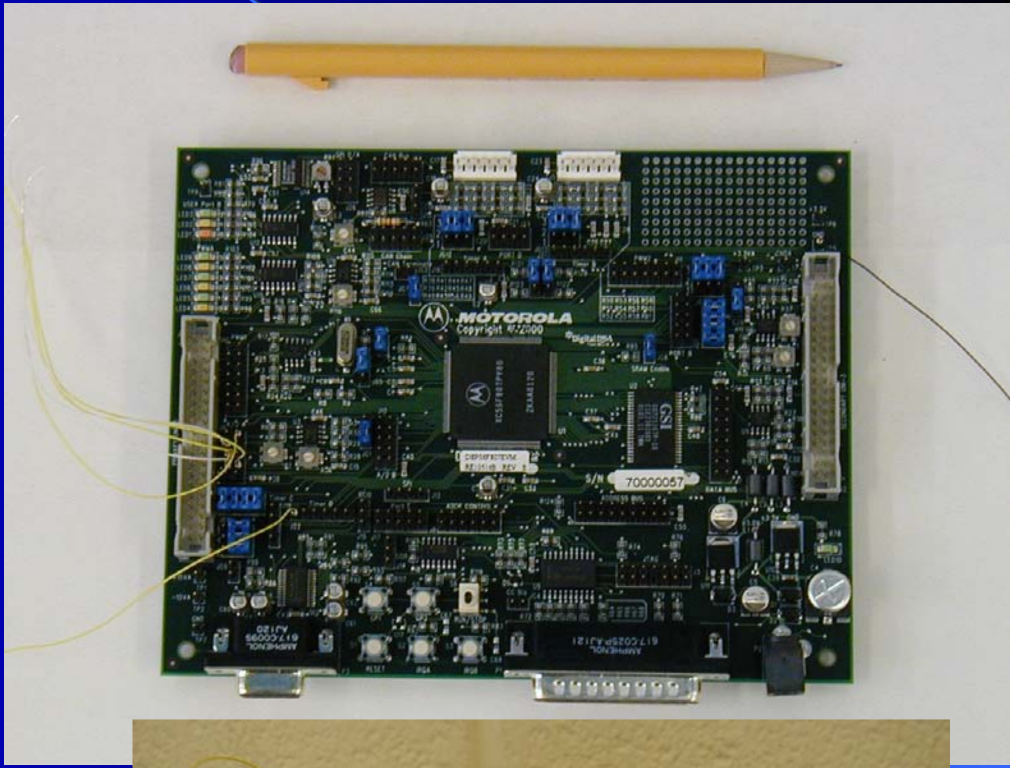
- Inexpensive, Almost-Ready-to-Fly
- Wingspan: 69.5”, Flying Weight: 8 lbs
- 2-stroke, glow engine

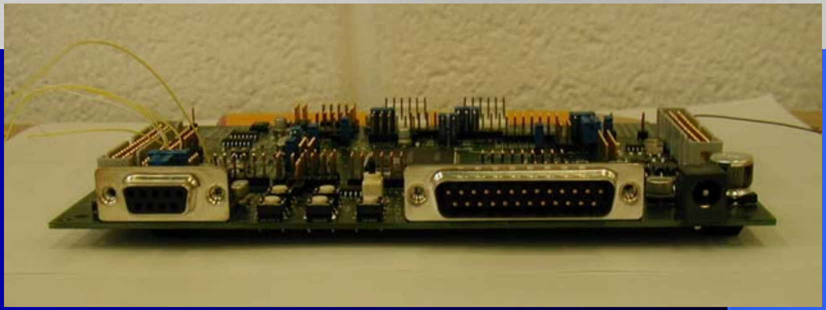
System Redesign



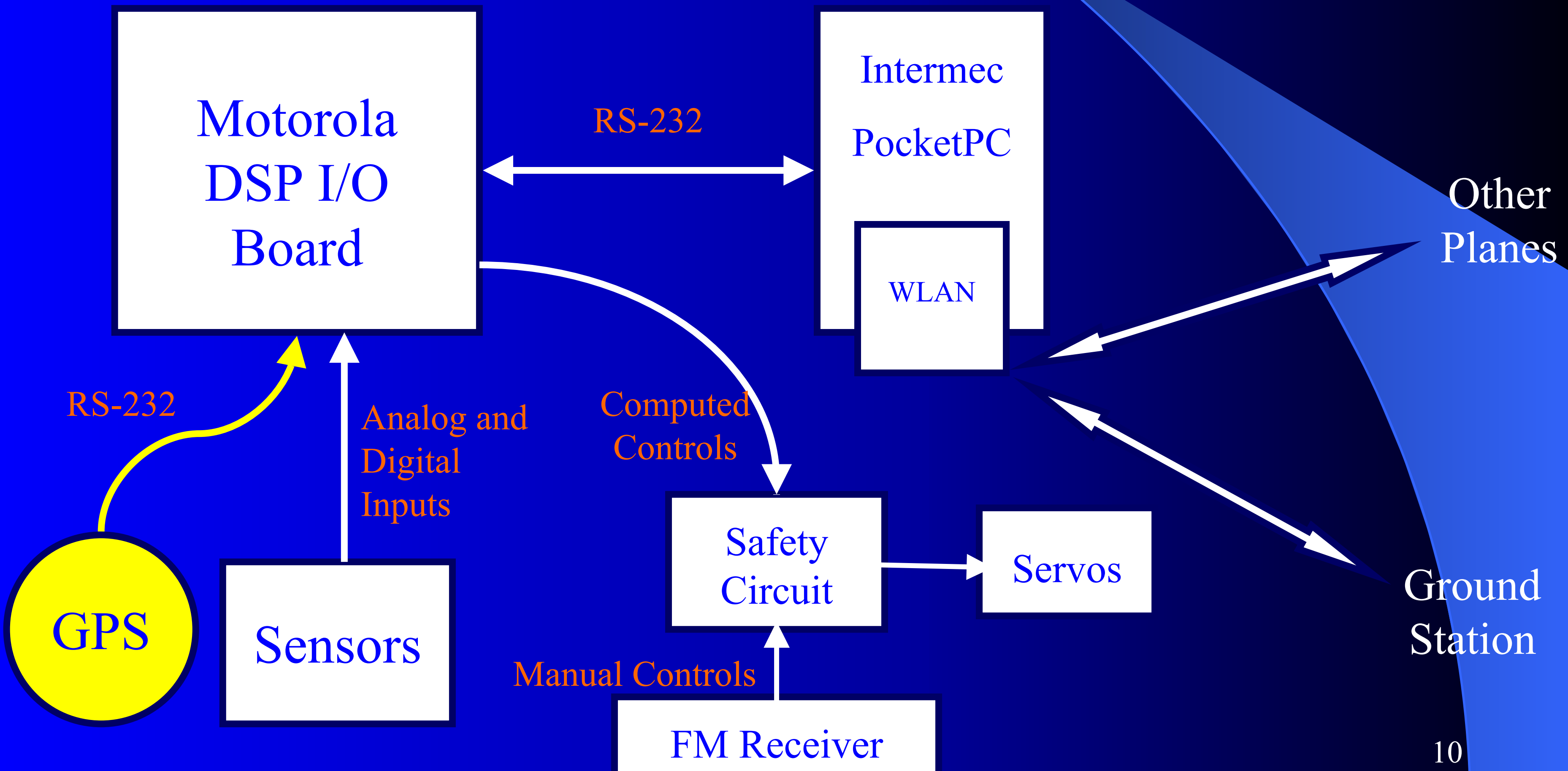
Motorola DSP56F807

- High performance DSP/Microcontroller with I/O board
  - Harvard-style architecture with up to 6 operations per instruction cycle
  - 80 MHz clock speed
  - 12 PWM channels, 32 digital I/O pins, 16 A/D pins
- Acts as I/O board and CPU
  - Reads pressure sensors, rate gyros, and accelerometers
  - Parses GPS input
  - Computes controls and outputs PWM to servos
  - Receives waypoints and commands from PocketPC and transmits status and position





System Redesign

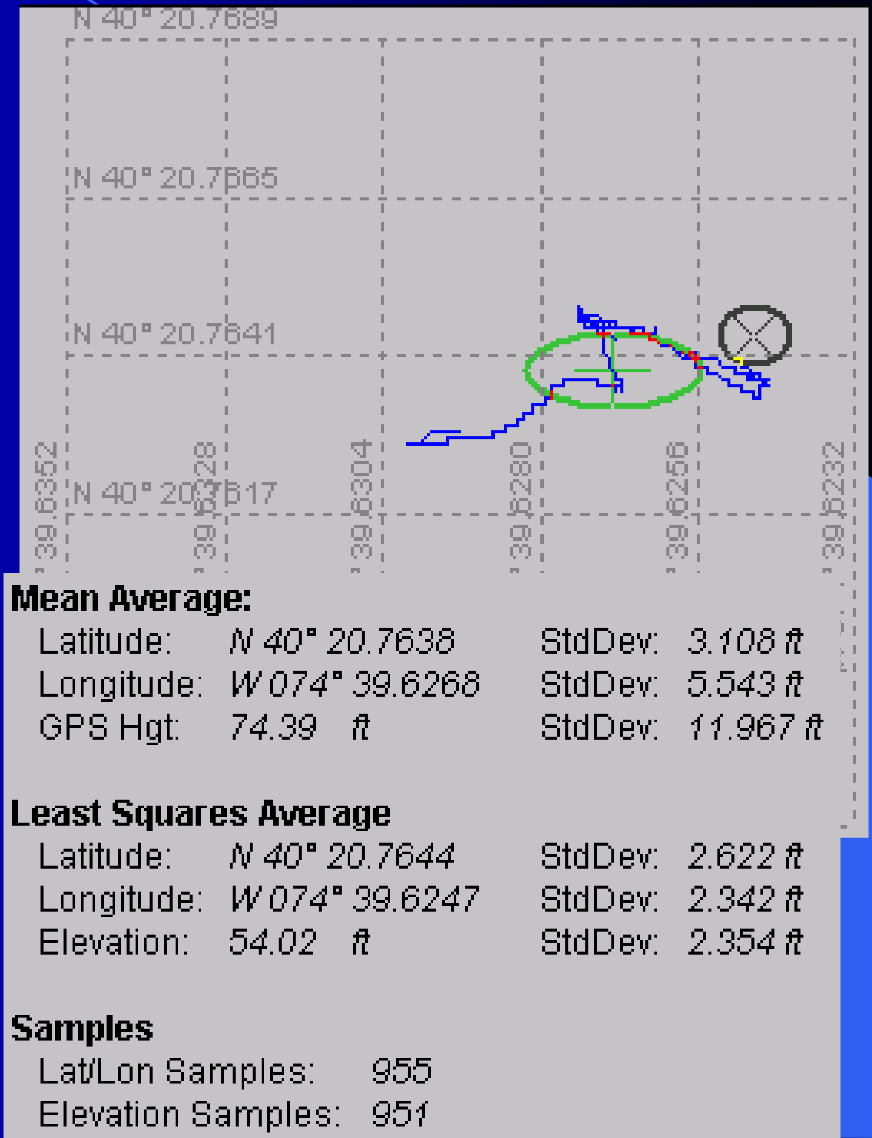


Pharos iGPS Receiver

- Sampling rate: 1 Hz
  - Used in conjunction with inertial sensors
- NMEA-0183 Protocol read serially by DSP board and parsed

GPS Accuracy vs. Precision

- Inexpensive receiver = terrible accuracy
  - 5 meter max accuracy
  - Not good enough for formation flight
- However, only precision really matters
  - Global 5 meter error insignificant
  - Experimental standard deviation at fixed location small



NMEA Protocol

```
$GPGGA,023125.999,4020.7736,N,07439.6275,W,1,03,3.2,-0.1,M,,,,0000*31
$GPGSA,A,2,04,07,24,,,,,,,,,,3.4,3.2,1.1*30
$GPGSV,2,1,06,16,68,107,,04,68,021,39,24,60,288,42,30,49,255,*78
$GPGSV,2,2,06,07,43,120,41,18,22,109,*78
$GPRMC,023125.999,A,4020.7736,N,07439.6275,W,0.07,138.47,251102,,*18
$GPGGA,023126.999,4020.7737,N,07439.6274,W,1,03,3.2,-0.1,M,,,,0000*32
$GPGSA,A,2,04,07,24,,,,,,,,,,3.4,3.2,1.1*30
$GPRMC,023126.999,A,4020.7737,N,07439.6274,W,0.09,140.58,251102,,*14
$GPGGA,023127.999,4020.7737,N,07439.6273,W,1,03,3.2,-0.1,M,,,,0000*34
$GPGSA,A,2,04,07,24,,,,,,,,,,3.4,3.2,1.1*30
$GPRMC,023127.999,A,4020.7737,N,07439.6273,W,0.07,139.18,251102,,*16
```

Packet Sent every second, contains…

- RMC - Recommended Minimum GPS Data
- GGA – Global Positioning Fix Data

RMC Datafield

$GPRMC,154232,A,2758.612,N,08210.515,W,085.4,084.4,230394,003.1,W

1 2 3 4 5 6 7 8 9 A B C

The fields are:

1 Sentence identifier
2 UTC time - the above means 15 hrs, 42 min, 32 sec
  The seconds part may have a decimal
3 Validity: A for good, V for no good
4 Latitude - the above means 27 degrees, 58.612 minutes
5 Hemisphere: N for NORTH, S for SOUTH
6 Longitude - the above means 082 degrees, 10.515 minutes
7 Hemisphere: W for WEST, E for EAST
8 Speed (knots)
9 Track - degrees true (Bearing)
A UTC date - the above means day 23, month 03, year 94
B Magnetic variation - degrees
C Variation direction - W for WEST (+), E for EAST (-) = N or S

GGA Datafield

$GPGGA,023658.999,4020.7690,N,07439.6275,W,1,04,13.7,80,M,,,,0000*04

1 2 3 4 5 6 7 8 9 10 11 12 13

1 = Sentence identifier

2 = UTC (coordinated universal time zone). UTC used be known as GMT.

3 = latitude of the GPS position fix

4 = Hemisphere: N for NORTH, S for SOUTH

5 = longitude of the GPS position fix

6 = Hemisphere: W for WEST, E for EAST

7 = quality of the GPS fix (1 = fix, but no differential correction)

8 = number of satellites being used

9 = horizontal dillution of precision

10 = GPS antenna altitude in meters

11 = meters

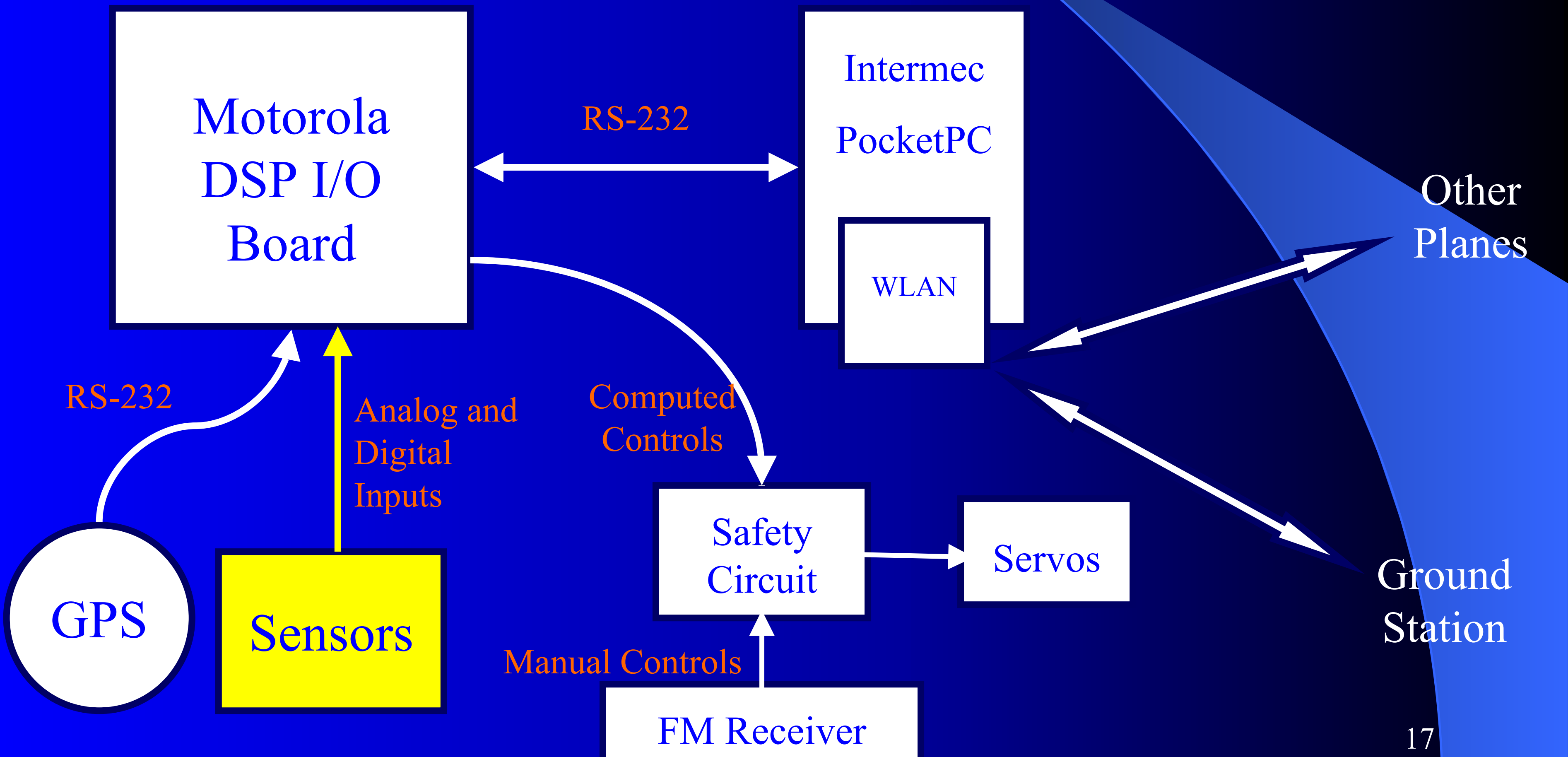
12 = deferential station's ID

13 = checksum for the sentence

GPS Parsing Algorithm

- DSP hardware interrupt fires after every 8 bits (1 ASCII character) received
  - Places data into circular buffer, marking the head and tail of the GPS sentence
  - Prevents loss of GPS data
  - Interrupt given priority over less important routines
- When sentence complete, buffer is parsed using Finite State Machine

System Redesign



Sensors

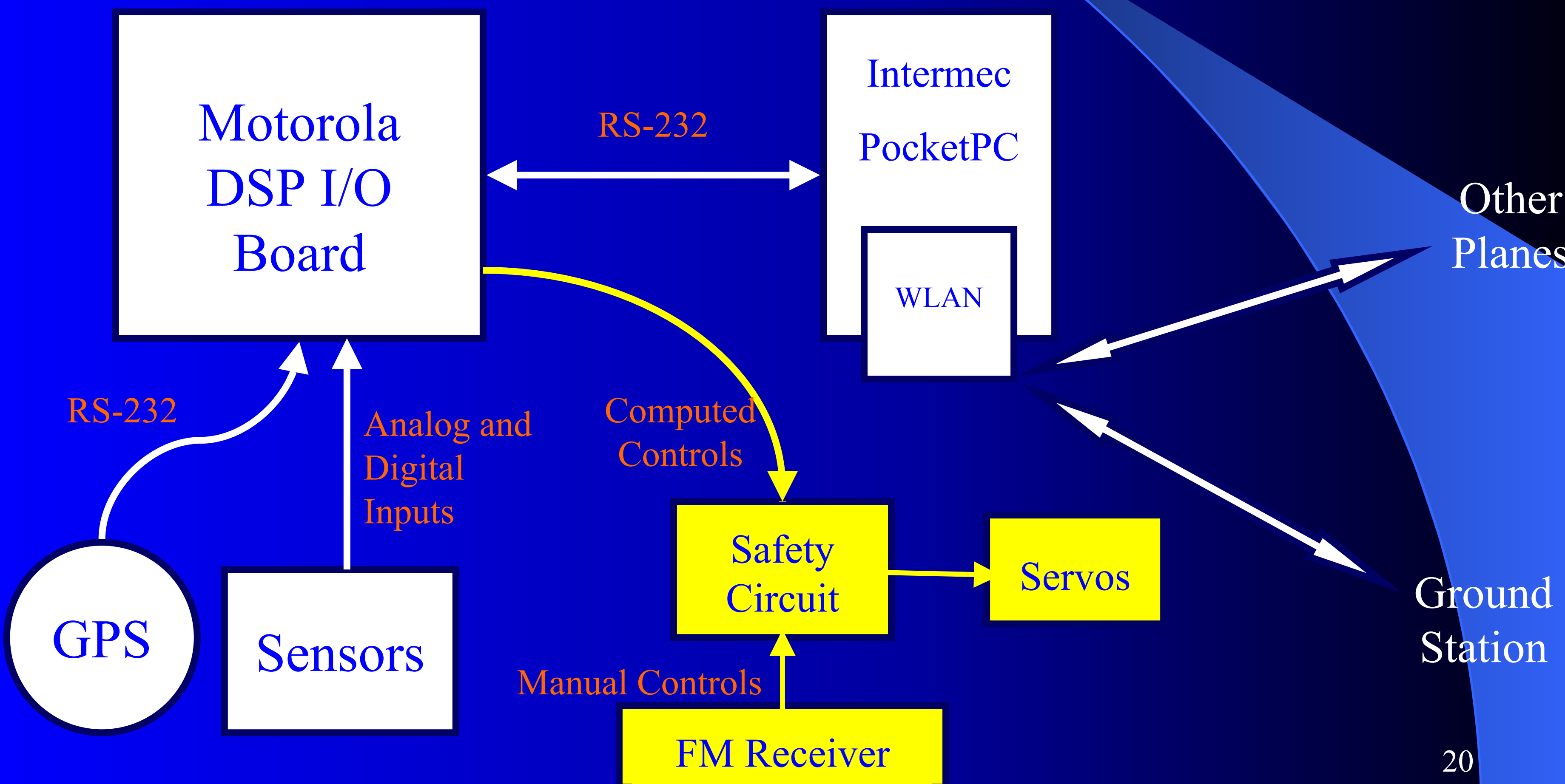
- Static and Dynamic Pressure Sensors
- 3-Axis Accelerometer
- 3-Axis Rate Gyros

Used for Inertial Navigation between GPS readings and system redundancy for failure tolerant control

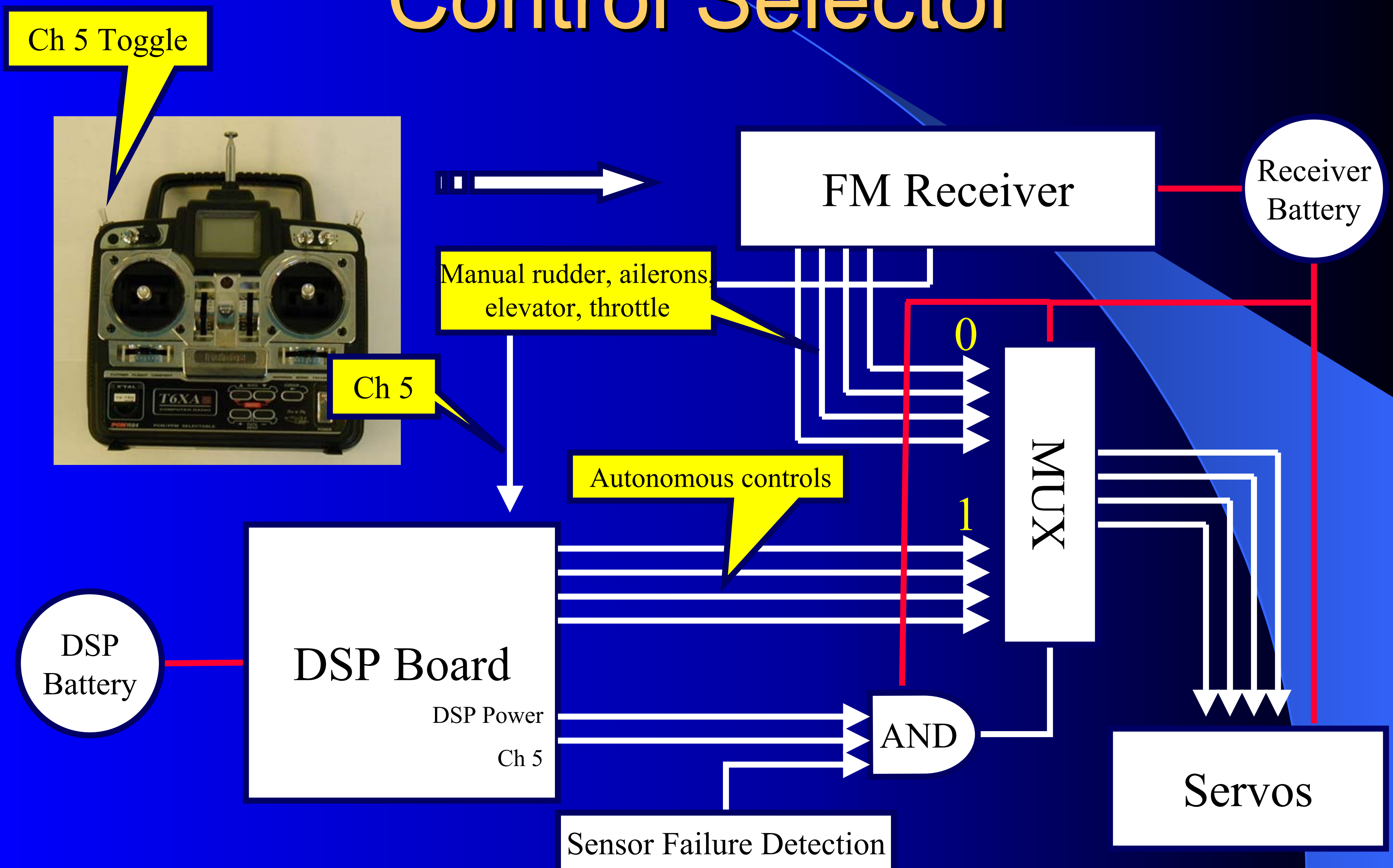
Failure-Tolerance

- Multiple sensors allow measurement redundancy:
  - Speed: GPS, air pressure sensors, integration of accelerometers
  - Position: GPS, double integration of accelerometers
  - Bearing: GPS, integration of rate gyros
- Two channel system considered *fail-safe*
  - Discrepancy between sensors indicates failed sensor – trips safety circuit
- Three channel system considered *fail-operational*
  - “Majority rules” identifies failed sensor

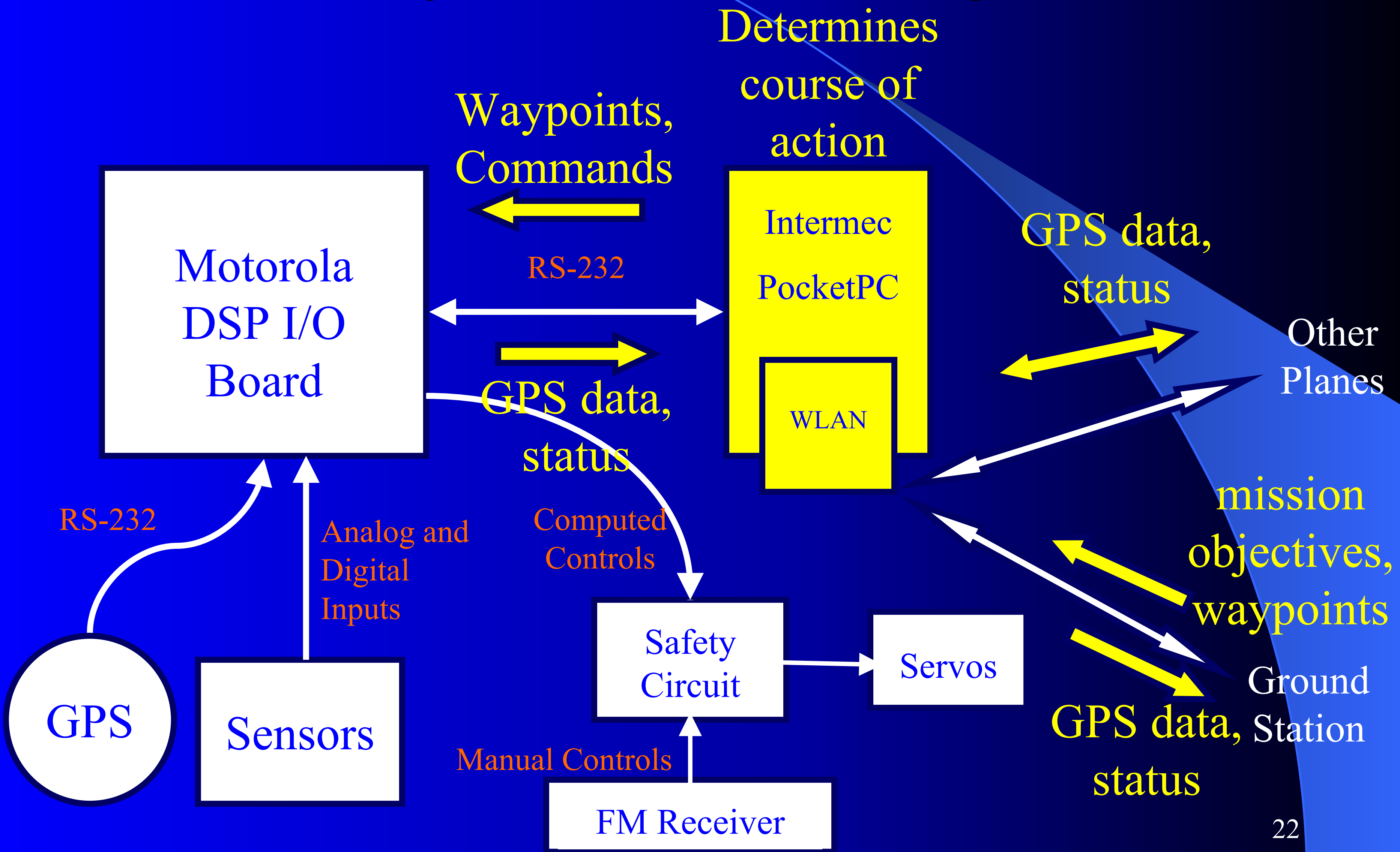
System Redesign



Safety System & Control Selector



System Redesign



Intermec 700 PocketPC

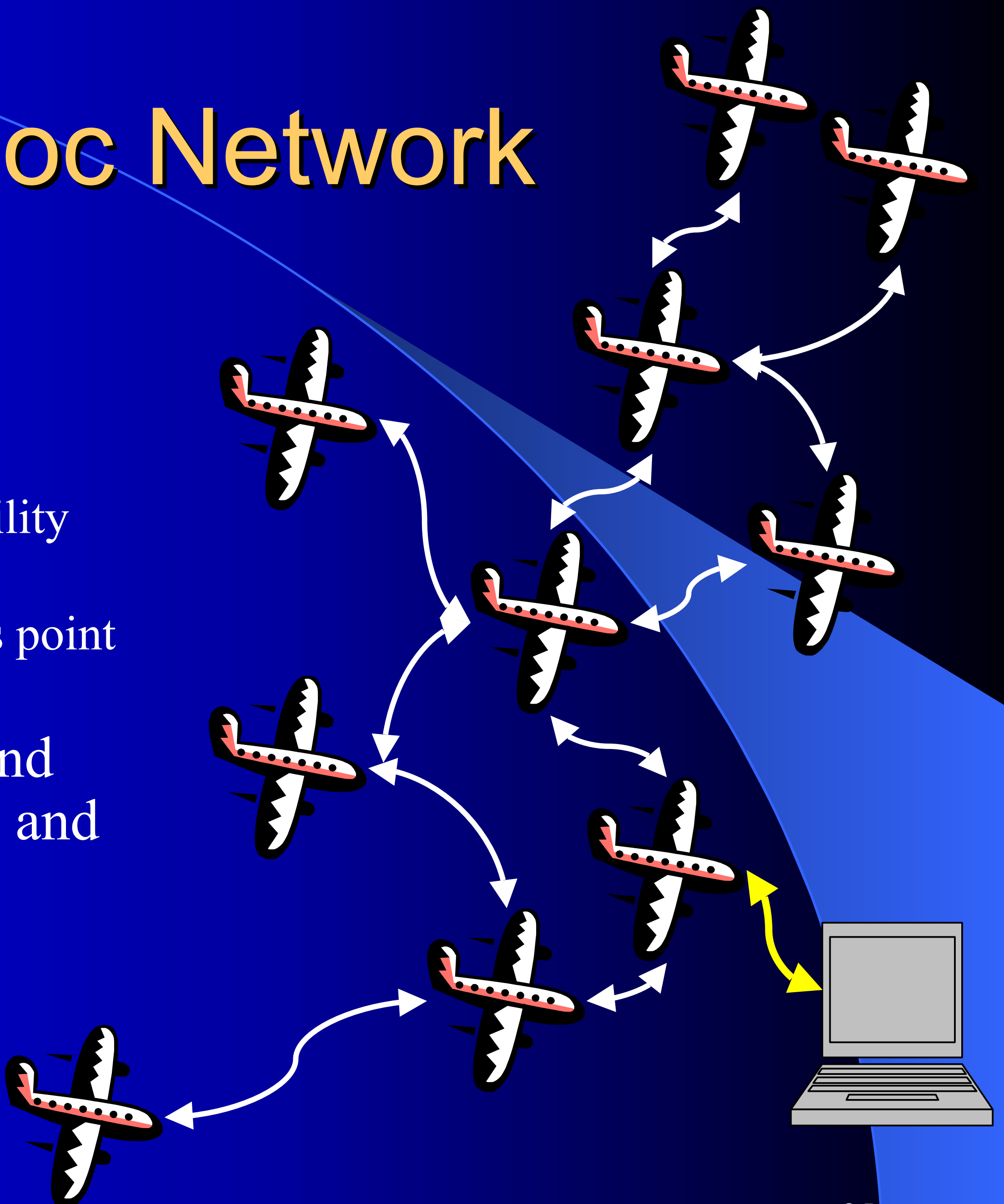
- 206 MHz Intel StrongARM processor running Windows PPC
- Integrated 802.11 WLAN (also Bluetooth and wireless modem)
- Rain, dust, and drop resistant

Wireless Network

- TCP/IP Sockets
  - Messages guaranteed to arrive
  - Error checking
- Communication link automatically re-established if plane goes out of range
- Each plane has its own IP address
- WLAN gives ~1800 ft. range

Ad Hoc Network

- Goal:
  - Extend range and communication reliability of each plane
  - Allow dynamic access point (mobile base station)
- Computation power and large memory of DSP and PocketPC allow data aggregation
- “Multi-hop” communications

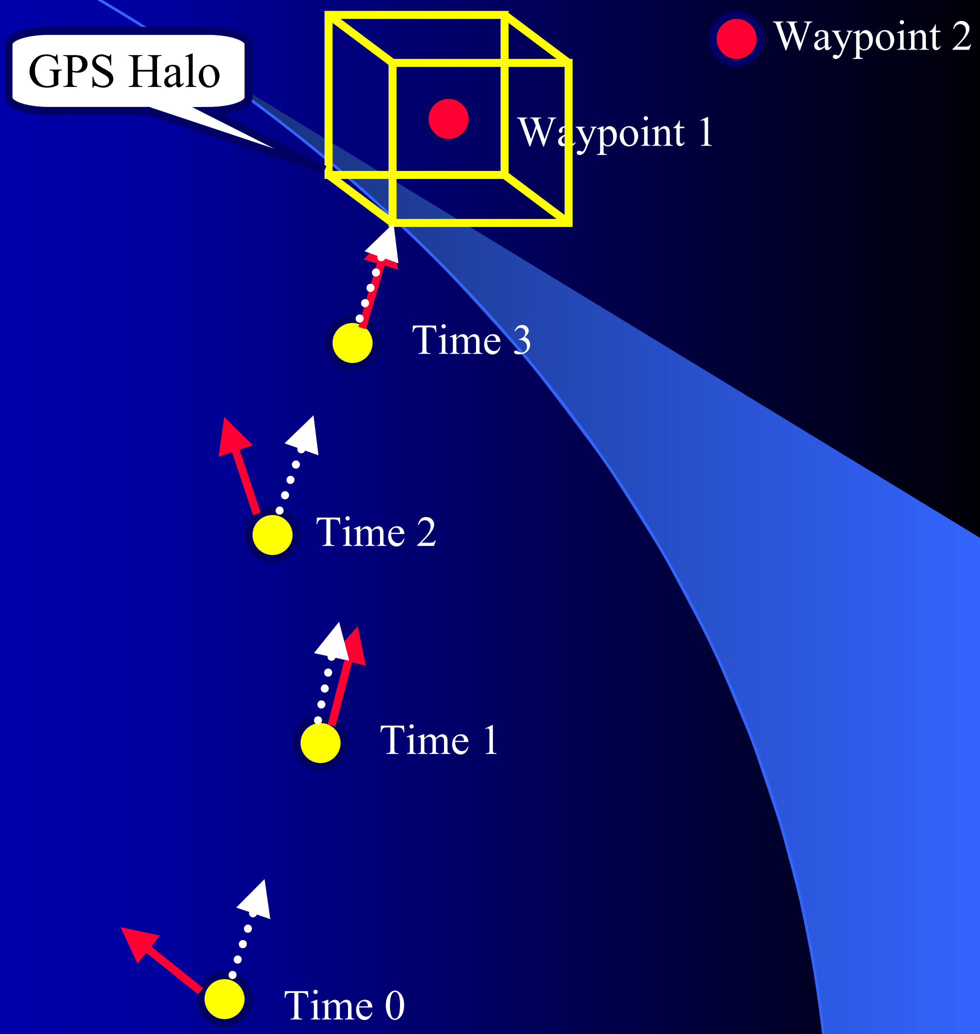


GPS Waypoint Navigation

- Calculate necessary bearing from current position and next waypoint:
  - Great Circle Distance (radians) = d = acos(sin(Lat1) sin(Lat2) + cos(Lat1) cos(Lat2) cos(Lon1-Lon2)
  - Classic bearing = acos((sin(Lat2) – sin(Lat1) cos(d)) / (cos(Lat1) sin(d)))
- Translate bearing to inner-loop control system
  - I.e., bank left, etc.
- Eventually extend to 3d

Waypoint Navigation (cont.)

- Current bearing and position updated every second
- Desired bearing calculated every second
- Discrepancy fed into control system
- PocketPC determines when waypoint reached and calculates bearing to next waypoint



Status and Future Work

Current Status:

- Inertial sensors integrated on DSP board
- GPS integrated with DSP board
- DSP control of actuators
- High-level waypoint navigation

What's next:

- Continued development of inner-loop control
- Implementation of failure-tolerant schemes
- Test flight of redesigned, single plane system

References

- Stengel, Robert F. http://www.princeton.edu/~stengel/Phoenix.html Accessed on 9/30/02.
- Stefan, Jeff. "Navigating with GPS". *Circuit Cellar*. Issue 123, October 2000.
- Anthony, Michael and Gerson, Chris. The Phoenix Project: Design and Flight Test of an Unmanned Air Vehicle. 2001.
- Horowitz and Hill. The Art of Electronics. Cambridge University Press, 1980.
- Stengel, Robert F. Intelligent Failure Tolerant Control. June 1991.